Conception et Implémentation d'un Toolkit Agent SyncML

Benjamin Zores

RAPPORT DE STAGE

Benjamin Zores

20 août 2003

Conception et Implémentation d'un Toolkit Agent SyncML

The MADYNES Research Team

Stage ESIAL - Deuxième Année
Année Universitaire 2002 - 2003

Rapport de Stage

Le stagiaire *Zores Benjamin*
demeurant au 12 Av. Jean Jaurès
54500 VANDOEUVRE

a effectué un stage d'élève ingénieur
dans le cadre de son enseignement de
deuxième année (2002-2003), dispensé à :

ECOLE SUPERIEURE D'INFORMATIQUE
ET APPLICATIONS DE LORRAINE
Boîte Postale 239
54506 VANDOEUVRE

au sein de l'entreprise d'accueil

LORIA-INRIA
Projet MADYNES
615 Rue du Jardin Botanique
54600 VILLERS-LES-NANCY

sous la tutelle de son maître de stage
Mr. STATE Radu

Table des matières

Avant Propos

Le présent stage en entreprise s'est déroulé sur une durée de 10 semaines, du 23 Juin 2003, au 29 Août 2003, au sein du Laboratoire lorrain de Recherche en Informatique et ses Applications (LORIA).

Ses objectifs étaient de "*permetttre la découverte et la pratique des techniques et outils utilisés dans les métiers de l'informatique et de la production industrielle et de confronter l'élève-ingénieur aux contraintes temporelles, économiques et humaines associées*".

Mes objectifs personnels de carrière n'étant pas encore définis, j'ai longuement hésité à suivre un enseignement alterné (ESIAL / DEA) au courant de ma troisième année d'études. J'ai finalement préféré réfléchir à cette question encore un an. Aussi ai-je décidé d'effectuer mon stage de deuxième année dans une équipe de recherche et ce, afin de découvrir le monde de la Recherche. Mon stage de fin d'étude de troisième se fera dans une entreprise à proprement parler, suite à quoi je pourrai décider de l'évolution de ma carrière.

Je tiens donc, par le biais de ce rapport, à remercier vivement Madame Hélène KIRCHNER, Directrice du LORIA/INRIA, ainsi que Monsieur Olivier FESTOR, responsable du projet de recherches Madynes, au sein duquel s'est déroulé ce stage.

Des remerciements plus particuliers sont adressés à mon maître de stage, Monsieur Radu STATE, pour l'aide qu'il m'a apporté tout au long de ces 10 semaines, et qui ont permis la concrétisation de mon projet.

1 Présentation de l'entreprise d'accueil

1.1 Présentation du LORIA

Le Laboratoire Lorrain de Recherche en Informatique et ses Applications (LORIA) est une unité mixte de recherche commune à 5 établissements :

- **CNRS** : Centre National de la Recherche Scientifique
- **INRIA** : Institut National de Recherche en Informatique et en Automatique
- **INPL** : Institut National Polytechnique de Lorraine
- **UHP** : Université Henri Poincaré, Nancy 1
- **Nancy 2** : Université de Nancy 2

La création du laboratoire a été officialisée le 19 décembre 1997 par la signature du contrat quadriennal avec le Ministère de l'Éducation Nationale, de la Recherche et de la Technologie et par une convention entre les cinq partenaires. Il succède ainsi au Centre de Recherche en Informatique de Nancy (CRIN), et aux équipes communes entre celui-ci et l'Unité de Recherche INRIA de Lorraine.

Dans le secteur des sciences et technologies de l'information et de la communication, le LORIA possède, à travers 25 équipes de recherche, cent cinquante chercheurs et une centaine de doctorants, des compétences reconnues dans des secteurs en pleine évolution et porteurs de développement économique potentiel.

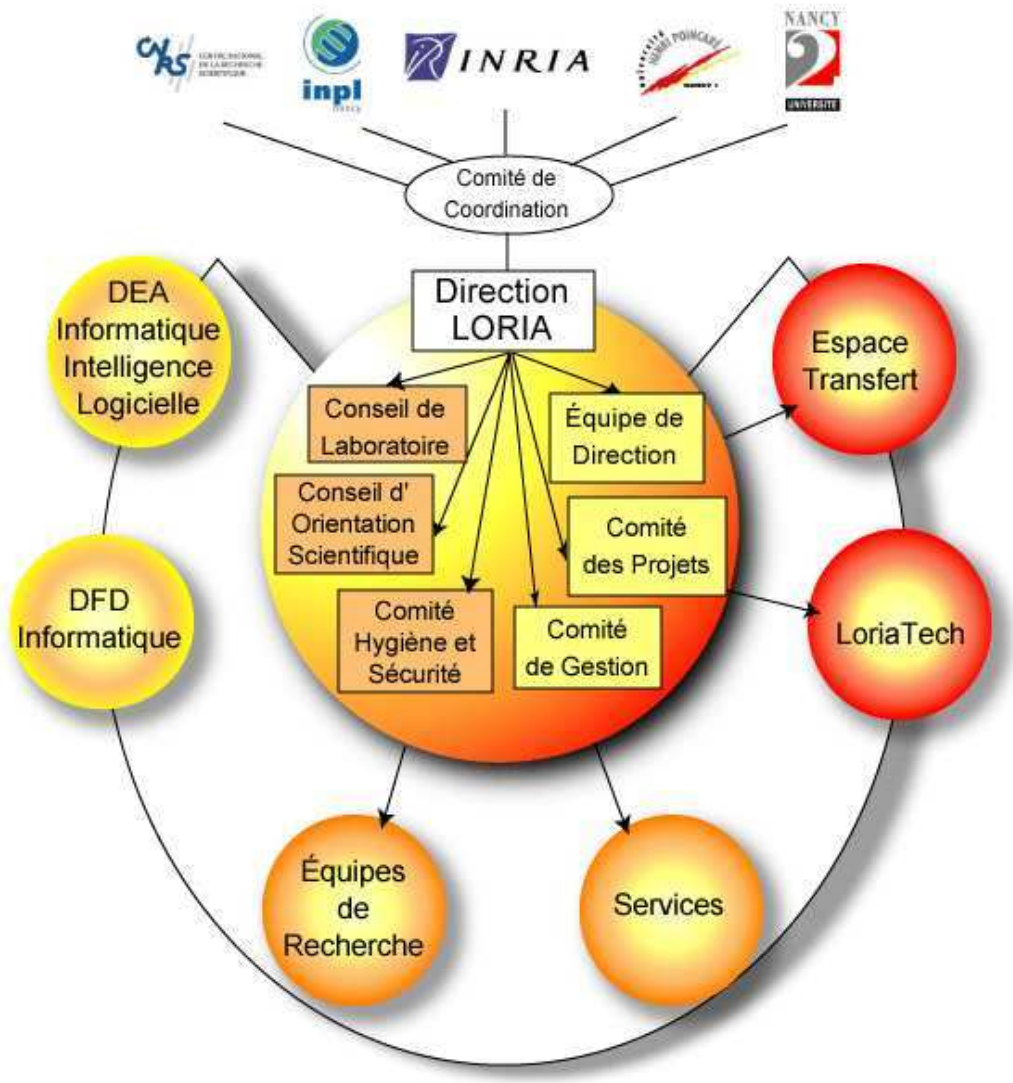
Les activités de ces équipes sont centrées autour de cinq thématiques principales "transversales" sur lesquelles elles développent des recherches fondamentales et appliquées. Les grands thèmes de recherche sont les suivants :

- Calculs, réseaux et graphismes à hautes performances
- Télé-opérations et assistants intelligents
- Ingénierie des langues, du document et de l'information scientifique et technique
- Qualité et sûreté des logiciels et systèmes informatiques
- Bioinformatique et applications à la génomique

1.2 Organigramme simplifié

Le laboratoire est sous la direction de Madame Hélène KIRCHNER depuis le 1er janvier 2001. Actuellement plus de trois cents personnes travaillent dans le laboratoire. Ces personnels sont répartis en 25 équipes de recherche et 8 services d'aide à la recherche.

L'organisation interne du laboratoire est relativement simple, comme le montre le diagramme suivant :



Chaque équipe rassemble des chercheurs, des doctorants et des assistants techniques ou administratifs, pour la réalisation d'un projet de recherche. L'ensemble des équipes de recherche est décrit ci-dessous :

- **ADAGE** : Algorithmique Discrète et ses Applications à la Génomique
- **ALGORILLE** : Algorithmes pour la Grille
- **CALLIGRAMME** : Logique linéaire, Réseaux de démonstration et Grammaires Catégorielles
- **CASSIS** : Combinaison d'Approches pour la Sécurité des Systèmes InfiniS
- **CORTEX** : Intelligence neuromimétique
- **DEDALE** : Développement de spécifications
- **ECOO** : Environnements pour la COOpération
- **ISA** : Vision artificielle et informatique graphique
- **LANGUE ET DIALOGUE** : Informatique linguistique pour le dialogue homme-machine multimodal
- **MACSI** : Modélisation, Analyse et Conduite des Systèmes Industriels

- **MADYNES** : Supervision des Réseaux et des Services Dynamiques
- **MAIA** : MAchine Intelligente Autonome
- **MERLIN** : Méthodes pour l'Ergonomie des Logiciels Interactifs
- **MIRO** : Systèmes à Objets, Types et Prototypes : Sémantique et Validation
- **MODBIO** : MODèles informatiques en BIOlogie moléculaire
- **MOSEL** : Méthodes formelles et applications
- **ORPAILLEUR** : Représentations de connaissances, raisonnements, et extraction de connaissances à partir de bases de données
- **PAROLE** : Analyse, Perception et Reconnaissance automatique de la parole
- **PROTHEO** : Contraintes, déductions automatiques et preuves de propriétés de logiciels
- **QGAR** : Querying Graphics through Analysis and Recognition
- **READ** : Reconnaissance de l'écriture et analyse de documents
- **SITE** : Modélisation et Développement de Systèmes d'Intelligence Économique
- **SPACES** : Systèmes Polynomiaux, Arithmétiques, Calculs Efficaces et Sûrs
- **TRIO** : Temps Réel et InterOpérabilité
- **TYPES** : Logique, Théorie de la Démonstration et Programmation

Afin d'assister la directrice et garantir la cohérence de la politique scientifique et le bon fonctionnement au quotidien, 5 instances ont été mises en place :

- **Équipe de Direction** : composée de plusieurs membres du laboratoire, elle assiste la Directrice dans ses fonctions
- **Comité de Gestion** : composé des chefs de service, il assiste le directeur dans le fonctionnement journalier du LORIA
- **Comité des Projets** : il conseille le Directeur sur la politique scientifique du LORIA, participe à l'évaluation des projets/équipes, et instruit les restructurations nécessaires
- **Conseil du LORIA** : il émet des avis sur la politique scientifique mise en oeuvre par le Comité des Projets. Sa composition est fixée par les statuts d'UMR.
- **Conseil des Orientations Scientifiques** : composé de représentants des équipes de recherche, il conseille la Direction dans la gestion scientifique du laboratoire

1.3 Personnel et Services

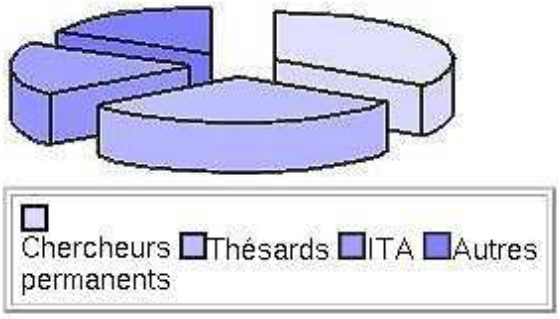
En Mai 2003, le LORIA regroupait plus de 300 personnes dont (les chiffres ci-dessus ne comprennent pas les projets Inria Messins et de l'IECN) :

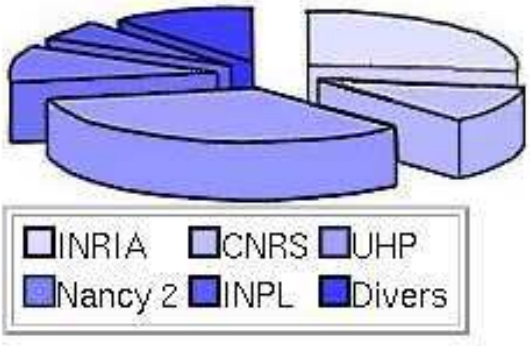
- Chercheurs permanents : 109
- Thésards : 94
- ITA : 61
- Post-Doc, ingénieurs experts, membres associés : 53

Ces personnels se répartissaient ainsi dans les établissements de rattachement :

- INRIA : 165
- CNRS : 43
- UHP : 64,5

- Nancy 2 : 40
- INPL : 24
- Divers : 73





1.4 Stratégie de développement

Le LORIA développe de nombreuses relations industrielles (plus de 200 contrats en cours fin 1999) et a une importante activité de diffusion de logiciels. Six logiciels et une marque ont été déposés en 1998. Fort de près de 90 enseignants-chercheurs, le LORIA participe activement à la formation universitaire dans les domaines des sciences et nouvelles technologies.

Conscient de la nécessité d'être un élément moteur du contexte socio-économique régional, le LORIA a décidé de créer le LoriaTech en janvier 1999, club des partenaires du LORIA, qui a pour objectifs :

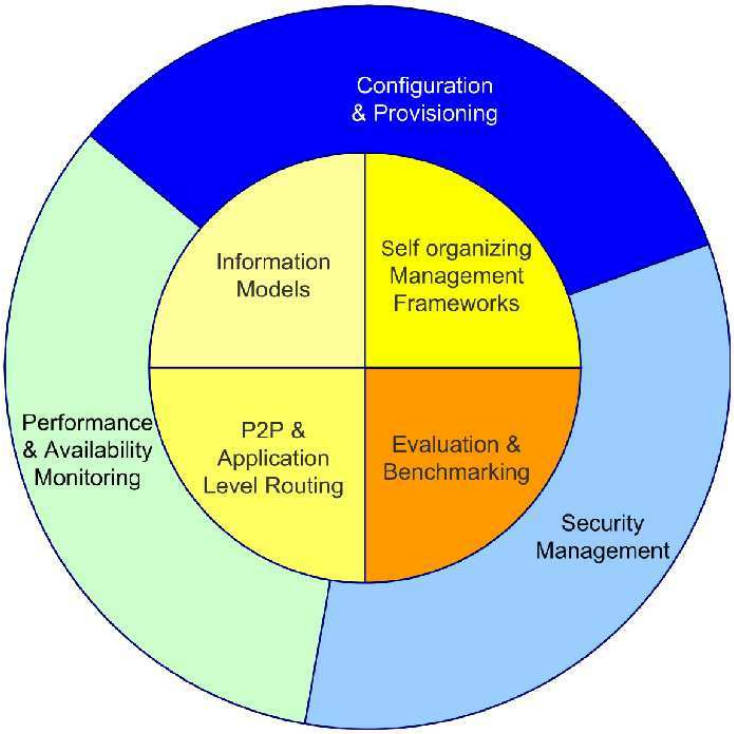
- donner accès plus rapidement aux informations sur les évolutions de la recherche et du secteur, c'est-à-dire améliorer la veille technologique des entreprises. En particulier, les membres du LoriaTech ont un accès privilégié au centre de documentation du LORIA et de l'INRIA-Lorraine.
- monter des partenariats divers, comme par exemple :
  - dans le cadre des établissements universitaires (ESIAL, ESSTIN, ISIAL, écoles d'ingénieurs de l'INPL, IUT Informatique de Nancy2, IUP-Miage de Nancy), la possibilité de stages de projets industriels.
  - dans le contexte du LORIA, les sujets DRT (Diplôme de Recherche Technologique), des conventions CIFRE.
- offrir la possibilité de partenariat LORIA-entreprises en réponse à des appels d'offre français ou européens.

1.5 L'équipe de recherche MADYNES

Nous nous intéressons désormais plus en détail sur l'équipe de recherche Madynes, qui m'a accueilli tout au long de ces 10 semaines de stage.

Le projet MADYNES (pour *Managing DYnamic NEtworks and Services* ou *Supervision des Réseaux et Services Dynamiques*)vise la conception, la validation et la mise en oeuvre de nouveaux paradigmes et architectures de supervision et de contrôle capables de maîtriser la dynamicité croissante des infrastructures et services de télécommunications et de résister au facteur d'échelle induit par l'Internet ubiquitaire.

Le diagramme ci-dessous met en avant les thèmes de recherches du groupe Madynes :



Le projet MADYNES s'organise selon 2 principaux axes de recherche :

- **Management autonome :**
  - conception de modèles et de méthodes permettant l'organisation autonome d'entités de management
  - conception et évaluation d'architectures de management, basées sur les systèmes Pair-à-Pair (P2P) ainsi que sur les principes de routage au niveau applicatif, selon une approche novatrice, permettant la représentation d'informations de management
  - modelisation et tests de performance des infrastructures et activités de management.

- **Zones fonctionnelles :**
  - *Sécurité :* nouveaux protocoles et infrastructures de cryptage par échange de clés pour des transactions confidentielles et respect de l'anonymay.
  - *Configuration et rovison de services :* automatisation des processus, allant de la souscription de services à leur déploiement et activation.
  - Mesures et analyses et instrumentation automatique des services proposés.

La seconde génération de l'Internet est le champ d'application principal du groupe. Son architecture et les services qui sont prévus d'être déployés offrent toutes les fonctionnalités, modulaires et dynamiques, auxquelles s"intéresse l'équipe MADYNES.

2 Déroulement du stage

2.1 Génèse du projet

La section ci-dessous a pour objectif de dresser un tour d'horizon des besoins technologiques actuels qui sont à l'origine du développement de ce projet.

2.1.1 Introduction

L'utilisation de périphériques portables reliés les uns aux autres a pris une place non négligeable dans la façon de vivre actuelle. On assiste en effet à une véritable prolifération des téléphones portables, assistants personnels et autres appareils sans fil, nécessitant de nouveaux moyens de gestion et de management.

Contrairement aux traditionnels équipements filaires, ces nouveaux périphériques doivent faire face à des contraintes économiques et technologiques nouvelles. En effet, les techniques de gestion utilisées se doivent de faire face aux moyens limités de bande-passante disponible, d'autonomie (gestion de l'énergie), aux problèmes de perte de signal et surtout à la nécessité que les informations contenues dans chaque appareil soient synchronisées.

Un scénario typique pourrait être un client possédant un périphérique portable. Un agent de management aura été implanté sur cet appareil, permettant sa gestion. Dans le cas où le client souhaiterait acheter un nouveau service pour son appareil auprès de son fournisseur de services, ce dernier se connecterait alors directement à l'agent de l'appareil client, synchronisant les informations possédées avec les nouvelles données du serveur. C'est ce que permet le protocole de communication SyncML.

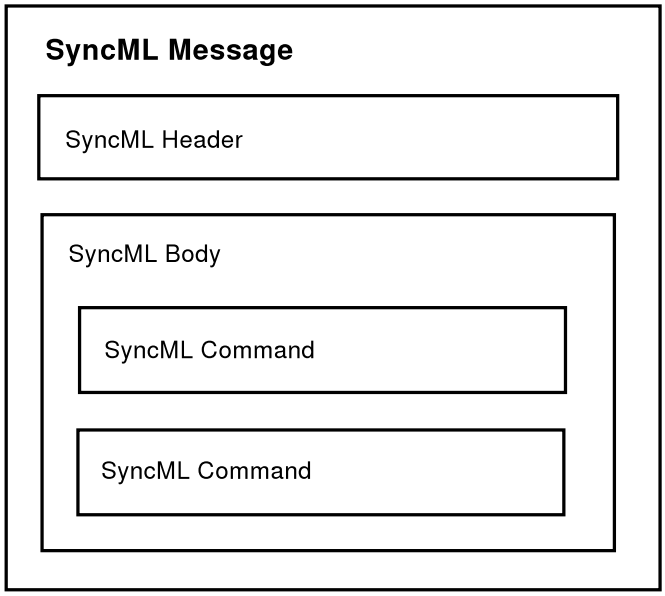
2.1.2 Le protocole SyncML

Il existe actuellement de nombreux protocoles de synchronisation propriétaires pour appareils mobiles. S'ils s'avèrent performants, ils n'en restent pas moins incompatibles entre eux et des appareils différents ne peuvent donc communiquer simplement. C'est dans le but de créer un protocole de synchronisation universel que des sociétés comme *IBM, Lotus, Motorola, Nokia, Ericson, Palm Inc., Psion et Starfish Software* ont fondé l'initiative SyncML.

Basé sur la norme XML, SyncML (*Synchronization Markup Language*) est un protocole visant à créer un standard ouvert pour un synchronisation universelle des données (agenda, messagerie, carnet d'adresses ...) entre terminaux éléctroniques (ordinateurs, assistants personnels, téléphones ...), quels que soient le système d'exploitation et les logiciels utilisés.

SyncML utilise la notion de message pour transmettre des informations. Ce dernier comporte une entête (*header*), comportant les principaux champs d'indentification et autres informations

de communications, ainsi qu'un corps de message (*body*) dans lequel sont contenus les commandes à effectuer. Le schéma ci-dessous résume cette idée.



2.1.3 La norme XML

La norme XML (*eXtended Markup Language*) sur laquelle se base SyncML, est un langage de balisage extensible et standardisé par le World Wide Web Consortium (W3C). Les balises du langage XML définissent la sémantique des données et non leur forme, contrairement à HTML par exemple, qui en est dérivé. Le code ci-dessous correspond à un exemple de message SyncML, en langage XML :

```
<?xml version="1.0" encoding="UTF-8"?>
<MgmtTree>
  <VerDTD>1.1.1</VerDTD>
  <Man>--The device manufacturer--</Man>
  <Mod>--The device model--</Mod>
  <Node>
    <NodeName>SyncML</NodeName>
    ...
  </Node>
</MgmtTree>
```

2.1.4 Impact escompté

Comme nous le verrons plus loin, le but de ce stage était donc la réalisation d'une implémentation libre du protocole SyncML, permettant le management de périphériques. Nous entendons ici par le terme "*libre*", le fait que le code source de l'application soit librement accessible et modifiable par quiconque. Comme aucune implémentation libre et conforme aux spécifications de l'initiative SyncML n'existait, il était important d'être les premiers à proposer une telle implémentation.

L'impact escompté est donc de fournir la première version fonctionnelle et exploitable d'un agent SyncML, facilement extensible, de manière à ce qu'un équipementier puisse y ajouter ses fonctionnalités propres avec un minimum d'efforts, tout en conservant la compatibilité de son application avec l'implémentation générique réalisée.

2.2 Environnement de travail

La partie ci-dessous consiste en une description rapide de l'environnement de travail auquel j'ai été confronté au courant de ce stage. Je ne m'attarderai guère sur ce point, de par son caractère classique, conforme à ce que l'on peut s'attendre d'un laboratoire de recherches en informatique.

2.2.1 Environnement matériel

Ce stage ne nécessitant pas de moyens informatiques particuliers ou exigeants, l'essentiel a été réalisé sur des stations de travail légères (de type Terminal X). Chaque poste permet une connexion sous un système GNU/Linux ou Microsoft Windows, où l'on retrouve les principaux outils logiciels de base. Seuls les tests applicatifs finaux ont été réalisés entre 2 ordinateurs portables, au moyen d'une connexion sans-fil de type Wifi.

2.2.2 Environnement logiciel

Ici encore, l'environnement logiciel est classique. L'implémentation s'étant faite en langage Java, j'ai décidé d'utiliser le compilateur Java de chez Sun Microsystems (JDK 1.4.2), permettant d'assurer une meilleure compatibilité et portabilité de l'application (ce qui s'est avéré être le cas, le programme fonctionnant aussi bien sous Windows que sur un système Unix).

Je tiens simplement à remercier Mr. Laurent Berroyer, pour son implémentation du protocole de représentation SyncML, sur lequel je me suis basé pour continuer l'implémentation du programme. Cette première pierre de l'édifice à permis un développement très rapide du reste de l'application.

2.3 Cahier des charges du projet

Nous allons ici détailler davantage les besoins applicatifs ainsi que les critères auxquels l'implémentation réalisée devait répondre, le but étant la conception d'un framework applicatif. L'implémentation se devait d'être pleinement fonctionnelle, à la fois exploitable et extensible par un fournisseur de services.

2.3.1 Etat des lieux

Il est souvent extrêmement difficile de concevoir un projet en partant de zéro, qui plus est si ce dernier doit être finalement extensible. De même, les différences sont plus que perceptibles entre la première approche théorique, lors de la phase de conception du projet et la version défnitive.

Il est vrai que l'implémentation d'un protocole de communication n'est guère chose facile. Contrairement à un programme traditionnel, si le forme du projet est relativement libre pour

l'auteur, il n'en est pas de même pour le fond. L'implémentation doit se faire avec rigueur de manière à ce que l'application soit conforme avec les spécifications qui ont été faites de la norme SyncML.

Aussi, il a été nécessaire de lire attentivement différentes publications sur les spécifications du protocole. Ceci peut être contraignant dans la mesure où les choix retenus dans le norme SyncML ne correspondent pas toujours à la vision des choses du développeur mais cela permet néanmoins un ancrage plus rapide et efficace dans le projet.

En outre, une première implémentation du protocole de représentation SyncML avait déjà été réalisée. Cela m'a permis une implémentation plus facile des protocoles de management et de communications.

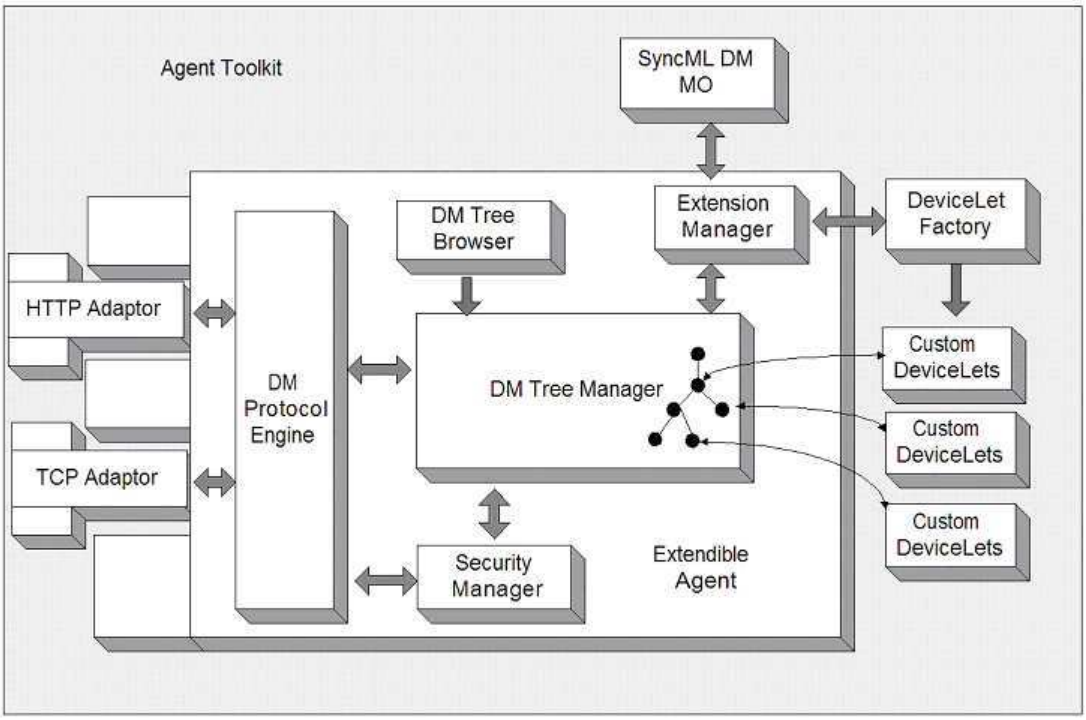
2.3.2 Phase de conception

Le projet d'implémentation SyncML se voit ramifié en de nombreuses sous-parties :

- **Representation Protocol** : outil permettant la création de messages au format XML
- **Management Toolkit** : ensemble d'outils permettant de générer des objets depuis un message XML et d'agir sur eux
- **Agent Manager** : agent permettant la communication entre périphériques ainsi que leur gestion

Suite à la phase de lecture des spécifications du protocole, la phase de conception a nécessité quelques jours. Durant cette phase, nous avons définis les différents packages et autres classes Java que nous utiliserons, ainsi que leur relations. Il est important de clarifier autant que possible le rôle de chaque classe afin d'arriver à un développement efficace.

La conception de l'application s'est faite sous Rational Rose, selon les techniques UML couramment utilisées. Le schéma ci-dessous permet une vue globale de l'architecture du logiciel. L'ensemble des schémas présentés ayant été réalisés de manière à être publiés internationnalement, vous excuserez l'emploi des termes anglais.

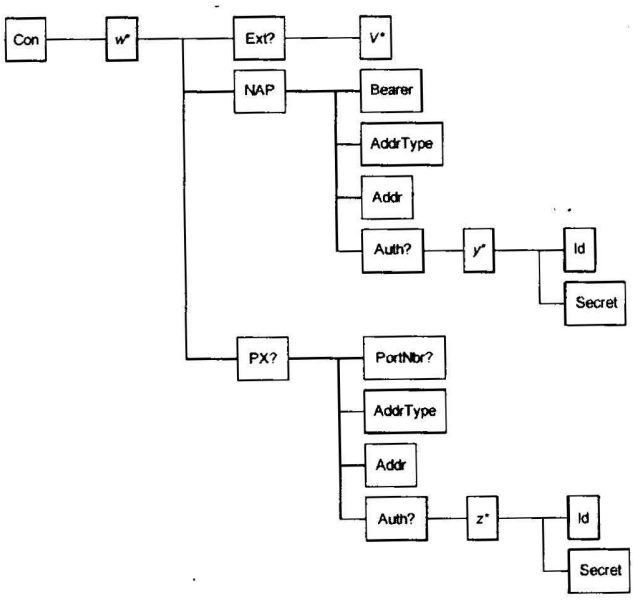


On y retrouve ainsi une architecture très modulaire, basée sur l'agent SyncML. Comme on peut le voir, ce dernier est facilement extensible, de manière à utiliser au choix différents connecteurs de communications (TCP / UDP / HTTP ...) ou Factory (unité responsable du traitement spécifiques des commandes reçues par l'agent).

2.3.3 Device Management Toolkit

Bien que le protocole SyncML utilise la langage XML pour communiquer, il est nécessaire que les informations contenues par un périphériques soient stockées. Un appareil peut contenir de nombreuses données comme par exemple le numéro de série, les identifiants de connexion (nom utilisateur, mot de passe, adresse du serveur ...) et bien d'autres encore.

Ces différentes informations sont stockées sous forme arborescente et regroupées sous le terme de *SyncML Device Management Object*. Chaque périphérique qui se veut compatible avec la norme SyncML se doit de posséder une représentation de 3 objets standardisés. Il est bien entendu possible d'en contenir de nombreux autres, qui peuvent être spécifiques à chaque équipementier, mais les 3 objets standardisés sont nécessaires pour permettre à des appareils de marque différente de communiquer entre eux.

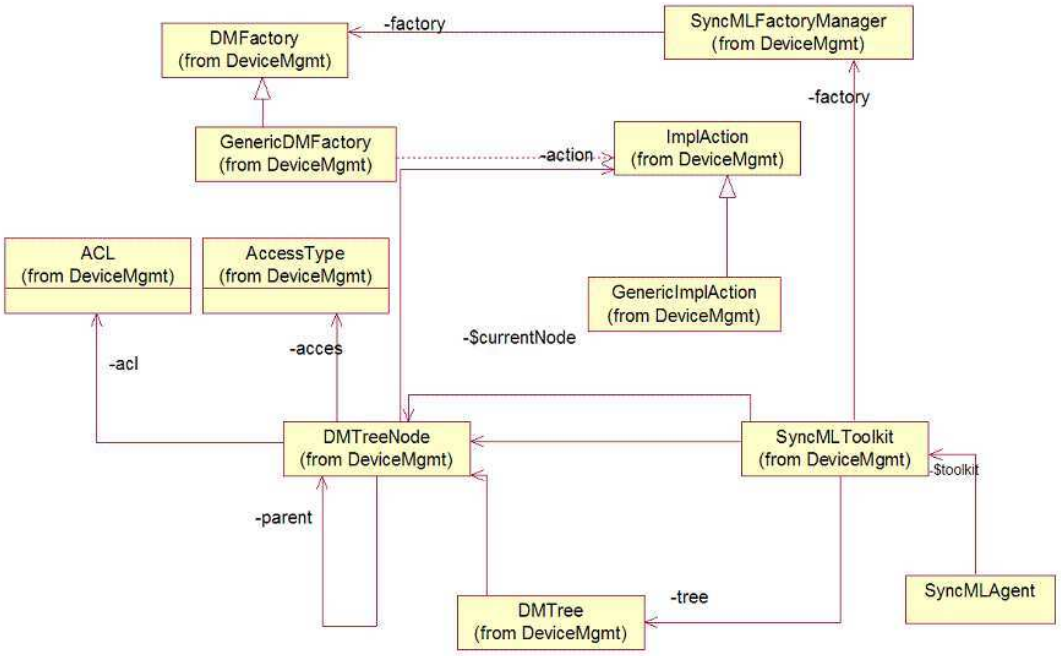


Le but de ce toolkit est donc d'assurer les opérations de management sur ces objets SyncML. La première des priorités était de réussir à recréer une structure objet en Java qui permette une représentation de ces informations de management. Les messages SyncML étant basés sur la norme XML, il est bien evidemment possible d'exprimer la configuration de l'arbre de management d'un périphérique auformat XML.

L'inverse se doit d'être également réalisable. Ainsi, ma première tâche était de concevoir un analyseur syntaxique conforme aux spécifications du langage SyncML. Cet analyseur, à qui l'on fournit en entrée un message XML, reconnaît les balises contenues pour créer des objets Java, correspondant à l'arbre de management ainsi qu'à autant de noeuds que nécessaires. Les objets sont liés entre eux en temps réel (association des liens de parenté) permettant de conserver la structure arborescente.

Le toolkit ainsi réalisé permet donc de créer un arbre de management pour un périphérique, d'y associer de nouveaux noeuds dynamiquement et bien entendu d'en supprimer. Une méthode de sérialisation a été mis en place, permettant de sauvegarder la configuration de l'arbre à l'extinction du périphérique et de la recharger en mémoire simplement.

Cet arbre de management contient donc l'ensemble des informations possédées par un périphérique. Les opérations de synchronisation ou de gestion en sont facilitée car il suffit pour un serveur de mettre à jour les branches de cet arbre ou de lui rajouter/supprimer des noeuds pour modifier les données. Le diagramme ci-dessous rend compte de l'organisation du Device Management Toolkit.



Les communications entre 2 périphériques se font selon le protocole SyncML, c'est-à-dire par transmission de messages au format XML. Typiquement, 6 actions de gestion peuvent être effectuées :

- **Add** : permet l'ajout d'un nouveau noeud de l'arbre.
- **Copy** : permet de copier la valeur d'un noeud dans un autre
- **Delete** : permet de supprimer un noeud et tous ses descendants.
- **Exec** : exécute un programme.
- **Get** : permet de récupérer la valeur contenue dans un noeud de l'arbre.
- **Replace** : remplace la valeur d'un noeud par une nouvelle.

A chaque noeud est associé un objet permettant de déterminer des actions spécifiques à effectuer dans le cas où des opérations de management auraient été effectuées. Ainsi, dans le cas où un serveur voudrait attribuer une nouvelle adresse réseau à un de ses clients, plutôt que de simplement modifier la valeur de cette adresse dans l'arbre de management du client, il peut être judicieux que l'agent contenu dans l'appareil client mette directement à jour son interface réseau avec la nouvelle valeur d'adresse via la fonction adéquate.

2.3.4 Couche de communication

Une couche abstraite de communication a ensuite été rajouté, permettant bien evidemment la communication entre périphériques. La volonté de modularité de l'implémentation globale du protocole, rend cette couche de communication relativement indépendante du protocole SyncML. Cela signifie qu'elle pourrait très facilement être utilisée dans une application tierce, tout comme il est possible de la changer si une plus performante voyait le jour.

Indépendante du protocole, son seul rôle consiste en l'émission et la réception de données selon des principes réseau bien connus. Son contenu même a été modularisé. Il est ainsi possible de l'étendre et surtout d'utiliser le protocole réseau de son choix. Différents modules de communication sont disponbiles comme TCP ou UDP. Il est également possible de spécifier des communications dans un protocole réseau au niveau applicatif (comme HTTP).

Cette couche de communication peut s'étendre à plusieurs autres protocoles et l'agent SyncML peut choisir d'utiliser le plus adéquat en fonction des périphériques reliés.

2.3.5 Architecture de sécurité

Si les possibilités et facilités de management offertes par le protocole SyncML peuvent sembler intéressantes, il ne faut pas négliger le problème de sécurité qui peut en découler. En effet, des moyens doivent être mis en oeuvre sous peine de voir son périphérique contrôlé par une personne ou entité non autorisée.

Il n'existe pas, à l'heure actuelle, dans le protocole SyncML, de véritable méthode de cryptographie qui assurerait une confidentialité des données. Néanmoins, l'initiative SyncML préconise l'emploi de la méthode de hachage MD5, permettant une identification relativement sûre de chacun des protagonistes. C'est cette technique qui a été retenue dans notre implémentation.

Pour rappel, l'authentification MD5 permet la génération d'un code unique de 128 bits, généré à partir d'une chaîne de caractères. Prenons une illustration simple d'application de la méthode MD5. Un serveur SyncML possède dans son arbre de management, une liste de clients susceptibles de se connecter à lui. Pour chaque client, il dispose d'un nom d'utilisateur ainsi que d'un mot de passe.

Un client désireux de se connecter au serveur, envoi le code MD5 de 128 bits correspondant à la concaténation de nom utilisateur et de son mot de passe. Le serveur recoit donc de sa part une signature de 128 bits. Il génère également une signature à partir de sa base de données de clients autorisés et dans le cas où la signature envoyée par le client est identique à celle regénérée, le serveur est assuré de l'identité du client.

Cette méthode a l'avantage de ne guère nécessiter de temps ni de puissance pour calculer la signature MD5 mais surtout, permet d'éviter que le nom utilisateur et le mot de passe associé circulent en clairs sur le réseau. Le hachage MD5 étant algorithmiquement irréversible, il ne peut être décodé pour découvrir les identifiants.

Cette technique permet donc d'éviter les attaques de type *Man-In-The-Middle*, en s'assurant de l'identité de chaque périphérique communiquant.

2.3.6 Device Agent Manager

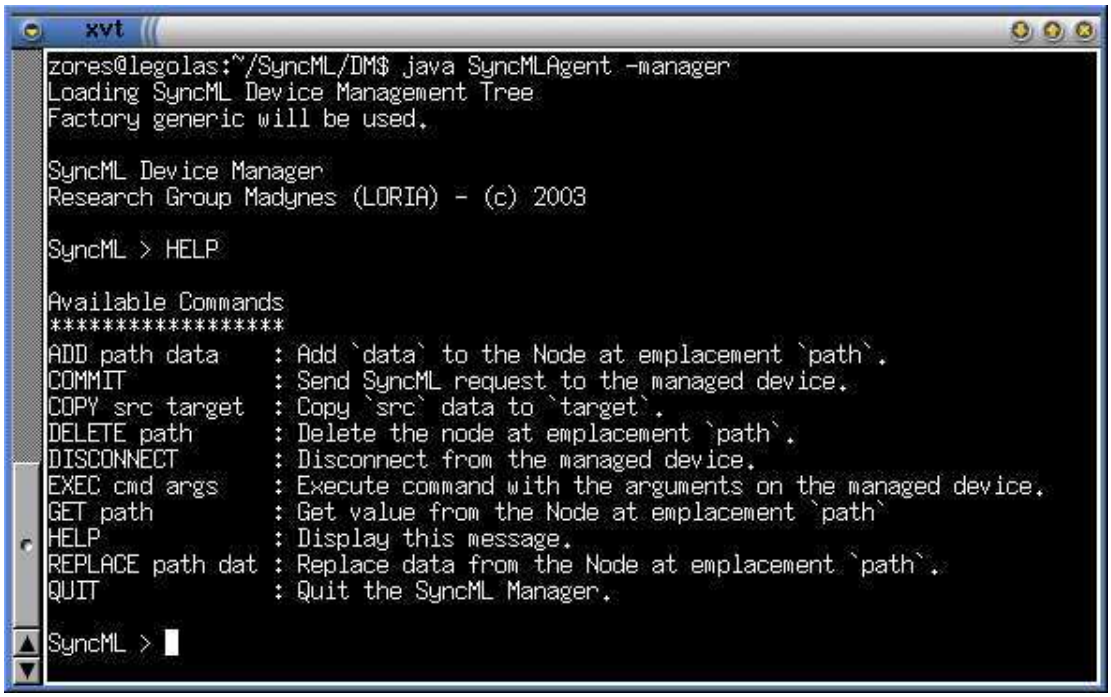
Le *Device Agent Manager* se veut être le coeur même du programme. Son rôle est de relier entre eux les différents modules en offrant une interface d'exploitation pour l'utilisateur. L'agent dispose de nombreux modes de fonctionnement mais c'est son aspect "manager" qui reste le plus intéressant. Sa grande particularité est qu'il peut être lancé indifférent sur un terminal en tant

que client ou sur un serveur SyncML dédié.

Parmi ses fonctionnalités classiques, on retrouve la possibilité de rajouter un noeud à l'arbre de management en parcourant un fichier XML, d'y supprimer des noeuds ou encore de restituer une représentation de l'arbre de management au format XML.

Il utilise également la couche de communication décrite précédemment ainsi que le module de sécurité pour dialoguer avec d'autres agents SyncML. L'échange d'informations entre 2 agents se fait tout naturellement via des messages XML, conforme aux spécifications du protocole SyncML. C'est à l'agent que revient le rôle de générer de tels messages et surtout d'en interpréter les réponses. Selon les réponses ou les requêtes reçues, il ordonnera l'éxecution de fonctions de management spécifiques à un constructeur ou à un noeud de l'arbre.

L'agent se veut donc extrêmement versatile et son mode le plus intéressant dans la gestion de périphériques distants réside dans son côté "manager".

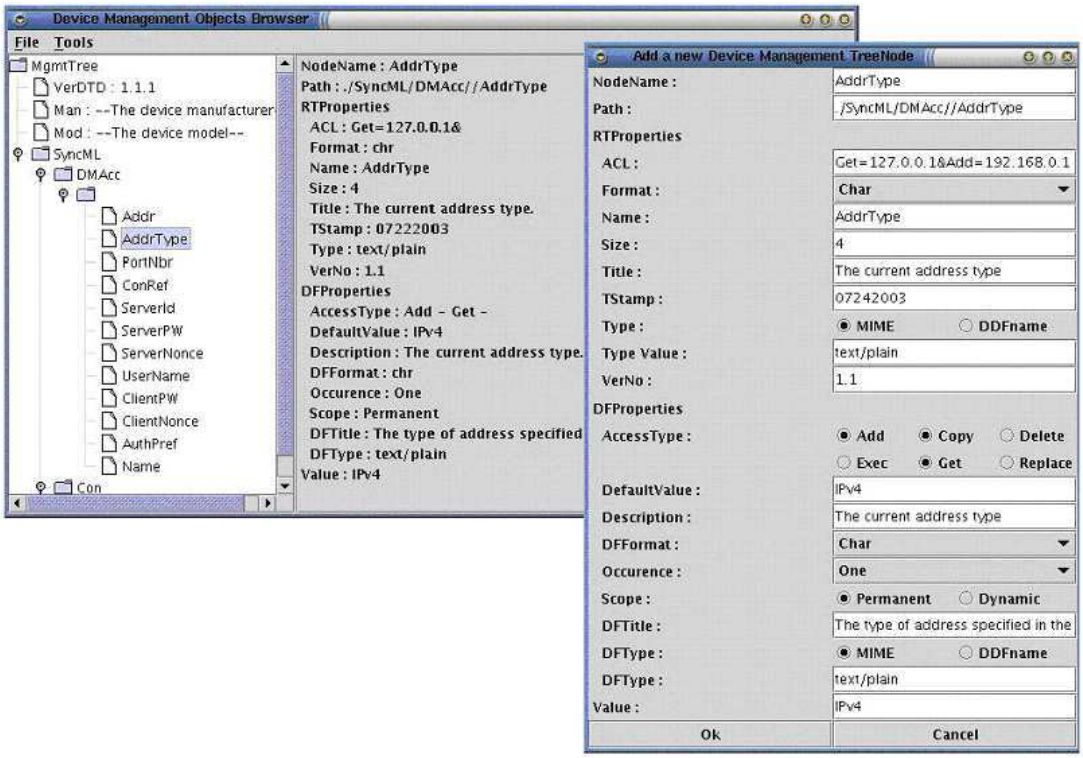


Il s'agit en fait d'une interface en ligne de commande qui permet la connexion à un agent distant ainsi que l'exécution des commandes de management vues précédemment. L'administrateur n'a alors qu'à saisir les requêtes qu'il souhaite voir effectuer sur l'appareil distant. Une fois la saisie terminée, un message de gestion SyncML est envoyé au destinataire qui traite la demande, par rapport à son arbre de management et renvoit les résultats au manager.

2.3.7 DM Objects Browser

Dans un soucis de commodité, il était également utile et agréable de de pouvoir visionner le contenu de l'arbre de management d'un périphérique au moyen d'un interface graphique.

Cette dernière est à double usage. D'une part, elle permet la visualisation rapide des champs et valeurs de chaque noeud composant l'arbre de management. D4autre part, elle permet d'ajouter très simplement de nouveaux noeuds à l'arbre ou encore d'en supprimer. C'est une alternative graphique et simplifiée à la saisie des noeuds en langage XML pour ensuite les rajouter via l'agent.



Cette interface présente néanmoins le défaut de ne pouvoir effectuer des modifications que de l'arbre de management local. L'ajout du support réseau permettant d'en faire un équivalent graphique à l'Agent Manager présenté ci-dessus semble donc être une bonne évolution possible.

2.3.8 Documentation utilisateur

Enfin, le projet n'aurait pas été complet sans la mise à disposition (aussi bien pour l'utilisateur final que pour le développeur) d'une documentation. Il était en effet important, pour que cette implémentation du protocole SyncML soit utilisable le plus rapidement et facilement possible, de fournir une documentation sur les principes de fonctionnement, à la fois de l'agent et de l'interface graphique.

Cette documentation, d'une vingtaine de pages, est disponible en anglais uniquement sur la page dédiée à l'implémentation SyncML du projet Madynes :

http ://www.madynes.org/software.html

Ceci conclut donc ce tour d'horizon rapide des objectifs que nous nous étions fixés au début de ce stage, ainsi que l'état actuel du projet.

2.4 Conception et Développement

Après cette description des tâches réalisées au courant de ces 2 mois de stage, nous allons à présent nous intéresser plus à la méthodologie employée pour y parvenir.

2.4.1 Méthode de travail

Le plus difficile lorsque l'on est assigné à une tâche nouvelle, c'est de savoir prendre ses marques. Il faut arriver à rentrer dans le projet qu'il nous est demandé de faire. Si le sujet peut se voir comme un projet classique de programmation, il n'en est rien.

En effet, ma connaissance du langage XML était jusqu'à présent, essentiellement théorique et celle du protocole SyncML était totalement inexistante. J'ai donc passé les premiers jours de mon stage à lire les différents articles publiés par l'initiative SyncML, faisant office de spécifications officielles du protocole. Cette phase était d'autant plus nécessaire que l'implémentation du protocole nécessite une certaine rigueur et laisse peu de place à l'improvisation ou à la liberté de mouvements.

Bien que cela ne soit pas particulièrement plaisant ni facile (les spécifications représentant pas loin de 250 pages, en anglais intégral), la compréhension des fondements du protocole est essentielle à un développement efficace. Il faut de même résister à la motivation du programmeur lancé sur un nouveau projet, à savoir la volonté de programmer au plus vite, pour prendre le temps de saisir les subtilités du protocole. Une fois ce stade de lecture achevé, le développement n'en est que facilité.

Il s'est ensuite suivi quelques jours dédiés à phase de conception sur papier de l'architecture globale de l'application. Nous avons, avec mon maître de stage, fait le point sur le rôle qu'aura chaque élément du programme et la façon dont le tout sera structuré. Ceci nous a permis de définir un squelette informatique sur lequel le développement pouvait se baser. A y réfléchir, il est intéressant de constater que même si, à l'heure actuelle, le programme reste conforme à la conception que nous avions réalisée, de nombreuses modifications ont été apportées.

Nous avons opté pour une certaine autonomie personnelle au niveau de l'implémentation à proprement parler, avec des entretiens fréquents. J'entends par là que je disposais d'une liberté totale pour implémenter le tout à ma guise, du moment que les spécifications du protocoles étaient respectées. Nous faisions le point sur l'état d'avancement des travaux tous les 2-3 jours. Cela permettait de clarifier la situation, mais également de définir le travail à accomplir pour les jours à venir.

Ainsi, nous nous sommes fixés des objectifs à atteindre à chaque semaine. De même, selon les ajouts de fonctionnalités, de nouvelles idées viennent à germer, comme l'interface graphique qui n'était originellement pas prévue mais qui, à y regarder de plus près, s'avère maintenant indispensable.

L'essentiel du *Device Management Toolkit* s'est vu terminé en environ 3 semaines. Nous avons ainsi pu dès à présent, utiliser l'outil afin d'effectuer toute une série de tests de performance. Cette première implémentation du toolkit ainsi que les tests effectués dessus, nous ont

permis l'écriture d'une publication scientifique, qui a dores et déjà été soumise à une conférence internationale prévue pour 2004.

Les semaines suivantes se sont déroulées de manière analogue, à savoir une implémentation libre de ma part, avec une supervision de la part de mon maître de stage, que j'allais régulièrement consulter en cas de problèmes ou tout simplement pour s'accorder sur les méthodes à employer pour arriver au résultat voulu.

2.4.2 Difficultées rencontrées et solutions adoptées

Il n'y a pas eu, à proprement parler de difficulté majeure, qui ait pu entraver le bon déroulement de ce stage. Bien evidemment, tout ne s'est pas fait tout seul et, avec plus de temps, certains choses mériteraient d'être faites différemment.

La principale difficulté reste néanmoins le fait de se plonger dans un domaine nouveau. Aussi bien les fondements de SyncML que l'utilisation pratique du langage XML nécessitent un certain temps d'adaptation (qui reste court cependant). De même, l'utilisation de stations de travail déportées si elles s'avèrent économiques, engendre une dépendance aux bons aléas du serveur. Une panne de ce dernier empêchant alors toute progression et la seule solution étant d'attendre.

Enfin, le contexte de programmation est assez différent de celui rencontré en milieu scolaire. Le code généré étant censé être accessible au plus grand nombre et surtout exploitable, il se doit d'être propre et stable. Il ne suffit pas que l'application fonctionne, il faut que le travail puisse être ré-utilisé par autrui.

Ceci implique une programmation avec rigueur, en choisissant des noms de variables qui ont un sens et qui permettent d'être rapidement comprises, de faire des commentaires de code pertinents (et en anglais uniquement) et d'éviter l'apparition de tout message d'erreur ou de débogguage non sollicité chez l'utilisateur final. Ceci implique en outre un choix judicieux de la license d'utilisation du code source produit.

3 Bilan

Pour ma part, le bilan de ce stage a été plus que positif, aussi bien au niveau professionnel que sur un plan plus personnel. Il a permis d'affiner mes connaissances techniques tout en éclaircissant mon choix de carrière.

3.1 Implémentation libre et fonctionnelle du protocole SyncML

Sur un plan strictement professionnel, ce stage aura permis la diffusion de la première implémentation libre du protocole SyncML. Avec l'avènement des nouveaux moyens de communications et principalement des technologies sans-fil, il n'est pas à douter que des protocoles de synchronisation et de management tels que SyncML rendront de grands services.

Il existe très probablement de nombreuses implémentations propriétaires de bien meilleure qualité que celle-ci chez les constructeurs (pour la raison évidente que les enjeux et moyens économiques sont autres) mais une telle implémentation open-source assure une compatibilité maximale entre des équipements différents et surtout, est librement utilisable et modifiable par quiconque. L'application actuelle a en effet été placée sous la license GPL (General Public License) qui implique qu'aucune utilisation des sources actuelles dans un projet propriétaires à sources fermées ne pourra être faite.

J'ai réellement apprécié travailler sur le développement de ce protocole, car je sais qu'avec les soutiens industriels que constitue l'initiative SyncML, il est destiné à devenir un standard dans les années à venir. Bien que de nombreuses choses restent encore à parfaire, je serai ravi que cette implémentation puisse être la base de futurs travaux et d'avoir fait parti des précurseur de la norme SyncML.

3.2 Publication scientifique IEEE

Le développement de ce projet aura également permis l'écriture d'une publication scientifique, qui a été soumise à la conférence **NOMS** pour 2004. Cet article, intitulé *An extensible Agent Toolkit for Device Management* a été co-écrit par M. Radu State, Olivier Festor et moi-même.

Il s'agissait pour moi de la première écriture officielle d'un article scientifique, qui plus est basée sur mes travaux. J'espère, aussi bien pour moi, que pour les membres de l'équipe Madynes, que cette publication sera validée.

3.3 Développement futur

Je n'ai aucun soucis à me faire concernant le développement futur du protocole SyncML, ses soutiens financiers étant assez représentatifs dans le monde des télécommunications. La tendance économique actuelle poussant de plus en plus d'entreprises à accuser leurs concurrents de monopôles et de concurrence déloyale, les tourne chaque jour un peu plus vers les logiciels libres.

Il n'est pas à douter que même si pour encore beaucoup, le logiciel libre constitue un échapatoire économique et juridique, la situation est en passe de changer. L'avenir est au libre et

de nombreuses entreprises l'ont parfaitement compris. Aussi, je ne peux que me réjouir personnellement, que l'on m'ait proposé de participer à cette implémentation open-source de SyncML.

Beaucoup de travail resterait néanmoins à faire sur l'implémentation qui a été réalisée. Je pense par exemple à une extension de l'interface graphique, de manière a permettre le management de périphériques distants tel qu'il est possible de le faire en ligne de commande via le manager. De même, de nombreuses fonctionnalités pourraient encore être rajoutées et un grand nombre de bugs doit encore exister mais, c'est à la fois le lot de tout programme, et les contraintes temporelles ne le permettent pas tout le toujours.

L'état actuel du programme reste néanmoins suffisamment avancé pour commencer à être exploité ou testé par les constructeurs, qui sont maintenant libres d'y ajouter leur propres fonctionnalités. Qui sait, l'avenir ne dépend que d'eux.

3.4 Bilan personnel

Sur un plan personnel, ce stage a été très bénéfique. Il m'a principalement permis de découvrir le monde de la recherche, qui m'a longtemps fasciné. J'y apprécie l'absence de contraintes économiques majeures comme on peut en trouver en entreprises. Les conditions de développement ne sont en effet pas soumises à une échéance temporelle crée par un service marketing et économique.

Techniquement, ce stage m'a permis de progresser dans la programmation en langage Java et de découvrir XML par la pratique. L'utilisation massive actuelle de ce langage de balisage lui confère en effet un avenir certain. Ce projet a également permis la conception complète d'une application, en partant persque de zéro (même si elle n'avait pas l'ampleur de grosses applications comemrciales).

La découverte de SyncML a pour moi été très bénéfique. J'ai à la fois pu découvrir un protocole qui sera peut être très utilisé dans un avenir proche et dont le développement réalisé m'ouvrira peut être des portes, et pris conscience de mon goût pour l'innovation. La publication scientifique qui a été co-écrite et soumise est également une grande opportunité pour moi, et j'espère vivement sa validation.

Si le monde de la recherche m'a jusque là interessé, c'est parce qu'il permet de travailler sur des technologies à venir. C'est ainsi à nous de faire en sorte que ces technologies deviennent des standards et s'imposent sur le marché. L'utilisation de technologies nouvelles suscite une motivation que l'on ne rencontre pas avec des projets plus ordinaires.

Ainsi, conformément à mes souhaits de départ, je compte rechercher un stage de troisième année dans une grande entreprise, plutôt que dans un laboratoire de recherche. Ma motivation actuelle me pousse à m'engager vers le service de *Recherche et Développement* de grandes entreprises, qui mettraient à me portée à la fois les technologies d'avenir et les moyens financiers permettant de les faire s'imposer.

Conclusion

Je tiens personnellement à remercier toutes les personnes de l'équipe Madynes qui ont, à la fois permis l'application de ce stage dans le cadre de ma formation d'ingénieur ESIAL, et son bon déroulement.

Ce stage m'a permis d'entrer de plein pied dans l'univers de la recherche en informatique et de répondre à grand nombre de mes interrogations sur mes orientations professionnelles. Il a également permis une implémentation pratique d'un protocole de communication que j'espère voir utiliser dans un avenir très proche.

Enfin, je souhaite remercier plus particulièrement Mr. Radu State pour son aide des plus précieuses au courant de ce stage et sans qui, cette implémentation n'aurait probablement pas vu le jour.

Annexes

Une description complète du code source de l'implémentation réalisée me semble à la fois trop complexe et sans grand intérêt pédagogique.

Aussi, ne figurera en annexes qu'un lien vers la page consacrée à SyncML sur le site Web de l'équipe Madynes :

http ://madynes.loria.fr/software.html

Vous trouverez sur cette page la version diffusée du Toolkit Agent SyncML ainsi que la documentation anglaise destinée à la fois aux utilisateurs de l'application et aux futurs développeurs.